UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

FUZZYSHARP TECHNOLOGIES INC.,

Plaintiff,

vs.

INTEL CORPORATION,

Defendant.

Case No.: 12-CV-04413 YGR

ORDER CONSTRUING CLAIM TERMS IN DISPUTE AND GRANTING SUMMARY JUDGMENT IN FAVOR OF DEFENDANT INTEL CORPORATION

Plaintiff Fuzzysharp Technologies, Inc. ("Fuzzysharp") brings this patent infringement action against Defendant Intel Corporation ("Intel") alleging that Intel HD Graphics Code "Sandy Bridge" infringes on U.S. Patent Nos. 6,172,679 ("the '679 Patent") and 6,618,047 ("the '047 Patent") owned by Fuzzysharp. Presently before the Court are two matters for decision: (1) the parties' claim construction dispute; and (2) the motion of Intel (Dkt. No. 28), originally filed as a motion for judgment on the pleadings, on the grounds that the asserted claims are not patent-eligible under 35 U.S.C. § 101 because the asserted claims of U.S. Patent No. 6,618,047 ("the '047 Patent") (claims 1, 8, 11, 12, 13, 46, 57, 64, 65, and 67) and of U.S. Patent No. 6,172,679 ("the '679 Patent") (claims 1, 4, and 5) (the "Asserted Claims")[1] are unpatentably abstract. The Court converted the motion to one for summary judgment, and permitted the parties to file supplemental briefing and evidence in support of and in opposition to the motion. (*See* Order Converting Motion for Judgment on the Pleadings Into Motion for Summary Judgment, Dkt. No. 45.)

Having carefully considered the papers submitted and the pleadings in this action, and for the reasons set forth below, the Court **ORDERS** as follows:

[1] At the time of the filing of Intel's Motion, there were 40 asserted claims. Fuzzysharp subsequently withdrew all but these thirteen. (Dkt. No. 66-2, Exh. B [Declaration of Victoria Q. Smith in, 8/9/2013 email from Dave Fink to Victoria Smith].)

(1) The remaining terms[2] for construction have the following meaning:

Term	**Construction**
Group 3('047 claim 1) "determining which of said at least one of 3-D surfaces or their sub-elements is always invisible or always visible"	"determining whether the 3-D surfaces are totally hidden or totally visible"
Group 4 ('679 claim 1) "determining for said viewpoint, whether each said selected surface is (a) an always unoccluded surface, an always hidden surface, or a remaining surface; or (b) an always unoccluded surface, or a remaining surface; or (c) an always hidden surface, or a remaining surface;"	"for each surface and from a predetermined viewpoint: (a) determining whether the surface is a totally visible surface, a totally hidden surface, or a surface that is neither totally visible or totally hidden; or (b) determining whether the surface is a totally visible surface or a surface that is neither totally visible or totally hidden; or (c) determining whether the surface is a totally hidden surface or a surface that is neither totally visible or totally hidden."

(2) The Asserted Claims are unpatentably abstract as a matter of law and Intel's Motion for Summary Judgment is **GRANTED**.

BACKGROUND

Fuzzysharp accuses Intel of infringing U.S. Patent Nos. 6,172,679 ("the '679 Patent") and 6,618,047 ("the '047 Patent"). The patents claim methods to reduce the number of visibility computations required to render a three-dimensional scene as a two-dimensional digital computer graphic image. The patents describe methods of analyzing a scene before employing prior art

[2] Over the course of the claim construction briefing, the parties agreed on constructions for all terms at issue except Group 3 and Group 4.

rendering techniques. In essence, the patent claims describe a method of analyzing which parts of a scene will be visible and which will be hidden, and which may or may not be hidden/visible depending upon the vantage point of the viewer.

I. THE TECHNOLOGY

In 3D computer graphics, computers generate two-dimensional images to represent three-dimensional scenes. In general, a three-dimensional scene is first established by specifying the objects in the scene, how they are lit, and from which point(s) and which angle(s) they will be viewed ("viewpoints"). Producing two-dimensional images of that three-dimensional scene requires the specification of a viewpoint and viewing direction relative to the scene. For a given viewpoint and in scenes with opaque objects only, at most one opaque object or surface can be visible along any single line of sight or viewpoint. For example, in a scene where a cube sits behind a sphere from the perspective of the simulated viewer, the sphere would block the viewer from "seeing" some or all of the cube. (*See* Figure 1.)

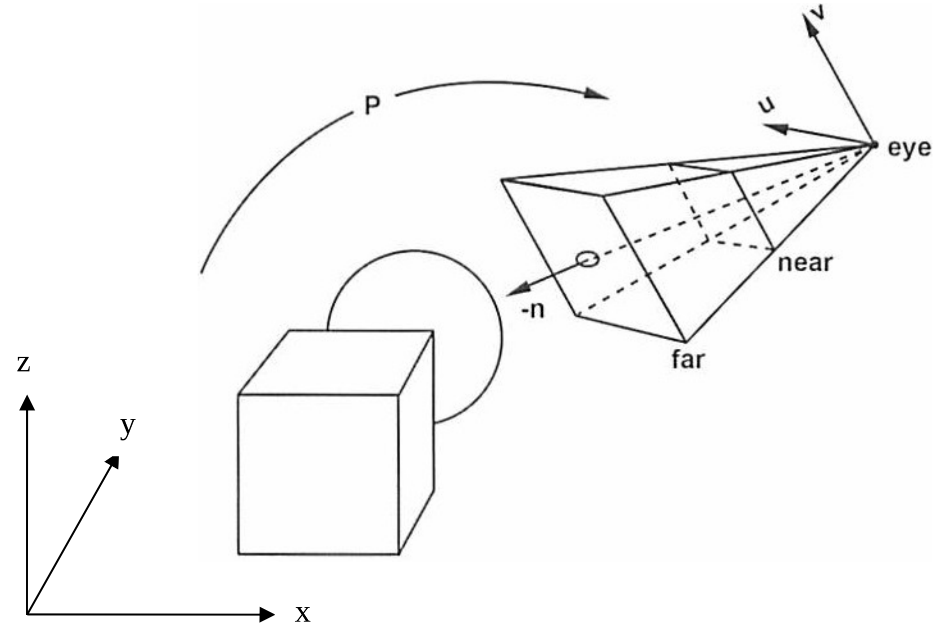

Figure 1: A simulated viewpoint (upper right) viewing a geometric scene model (lower left)

Generally the next steps in rendering a three-dimensional scene as a two-dimensional image are that the properties of the display device are specified, and the coordinates for display of the image on the screen of the device, *i.e.* the "pixels" (points of light with properties that may include color and intensity). The display screen contains a two dimensional array of pixels, each of which displays a color computed for some point in the scene. The pixels collectively form the images displayed on the screen. In order to determine which surface should appear in an image from a given viewpoint, the relative "depth" in the scene (*i.e.*, location on the z-axis) for the various object surfaces must be determined and compared. If objects are opaque and aligned such that one object "blocks" the view of another, then the object that is "nearer" to the viewpoint (*i.e.*, less "depth" in the scene) will have visible surfaces, while the object that is behind the first and "farther" from the viewpoint (*i.e.*, greater "depth" in the scene) will have surfaces hidden by the first.

3D graphics are typically rendered using GPUs (graphics processing units). The surfaces that will appear in the image from a particular viewpoint are represented as a collection of basic geometric shapes, or "primitives." (*See* Figure 2.) A GPU is specialized hardware that receives the primitives or surfaces from an application program and then displays the primitives or surfaces as an image on the computer screen. Upon receiving a primitive, the GPU converts it into a group of pixels that represents the points on a display device.

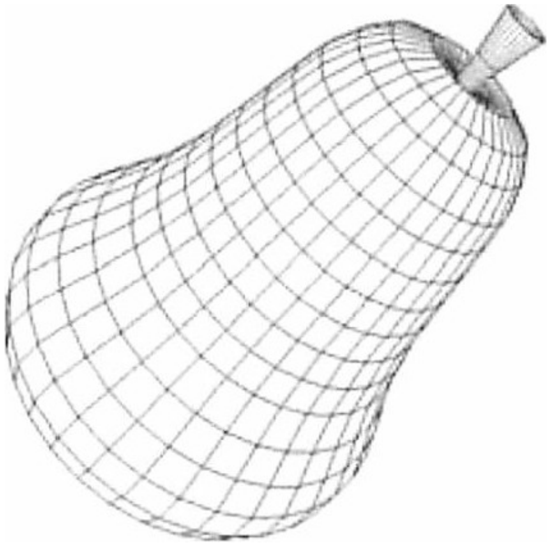

Figure 2: Collection of Primitives For Displaying Visible Surfaces of an Object From a Particular Viewpoint

GPUs typically use frame buffers, which are a portion of computer memory (buffer) organized into a discrete grid (frame) of pixels that store information about each pixel. In the frame buffer, each pixel has memory bits dedicated to representing the color and light intensity to be displayed at the location corresponding to that pixel. The frame buffer is connected to specialized computer hardware that can rapidly scan out the contents of memory to determine what should be displayed.

Many computer graphics rendering systems also include a specialized portion of computer memory designated as a "z-buffer" or "depth buffer." The GPUs can use the z-buffer to store data about the "depth" for each pixel for the purpose of eliminating hidden surfaces at the level of individual pixels or for groups of pixels. The process of using the z-buffer to determine which scene points are visible is known as "z-buffering" or "z-testing." For scene points that are determined to be visible, the values for color and other properties will be written into the frame buffer and become the values for the corresponding pixel. For a scene point that is determined to be hidden, the values for the color and other properties of the point will not be written into the frame buffer, and the data for the pixel that corresponds to that point will not be overwritten. Thus, the frame buffer and z-buffer are used together to determine what should be displayed on the screen.

II. The Patents

The patents-in-suit involve 3D computer graphics and purport to disclose "an improved method for performing visibility calculations." ('679 Patent Title, Field of Invention, col. 2:17- 20.) The patents describe the background art, and the problem to be solved by the patents, in part as follows:

> Visible surface detection is one of the most basic operations in 3D graphics. It is applied to generate images of surfaces directly visible to a viewer. . . . The standard strategy of visible surface detection is to divide surfaces into patch elements and compare the spatial relationship between these elements. Using this strategy, the visibility of surfaces cannot be determined until they have been analysed in detail. Although many techniques have been developed to address this issue, none are ideal as they either still require elaborate analysis of surfaces, or they impose various restrictions to the scene. The limitations of the current techniques can seriously affect the speed of visible surface computation. If the scene is complicated, many surfaces may be invisible. However, the image generation is often slowed down by the need to analyse each surface element in detail.

('047 Patent, col. 1:18-36.)

The ‘047 Patent is a continuation of the ‘679 Patent and shares a nearly identical specification. The only difference between the patents lies in the Summary of the Invention, which was modified in the ‘047 Patent to reflect differences in the claims in the patents. The Asserted Claims (claims 1, 4, and 5 of the ‘679 Patent, and claims 1, 8, 11, 12, 13, 46, 57, 64, 65, and 67 of the ‘047 Patent) describe a method (and variations) for reducing the number of visibility computations a GPU is required to perform in order to render a three-dimensional computer graphic image. The method reduces computations by pre-processing the surfaces from a group of viewpoints by: predetermining the visibility of surfaces from a group of viewpoints; creating data structures storing the resulting visibility computations; and then ignoring the pre-identified surfaces during later visibility computations, such as the visibility computations performed during rendering. Claim 1 of the ‘047 Patent is illustrative:

> 1. A method of reducing the visibility related computations in 3-D computer graphics, the visibility related computations being performed on 3-D surfaces or their sub-elements, or a selected set of both, the method comprising:
> identifying grid cells which are under or related to the projections or extents of projections associated with at least one of said 3-D surfaces or their sub-elements;
> comparing data associated with said at least one of 3-D surfaces or their sub-elements with stored data associated with the grid cells;
> determining which of said at least one of 3-D surfaces or their sub-elements is always invisible or always visible to a viewpoint or a group of viewpoints by projection based computations prior to a visibility computation; and
> ignoring said determined at least one of the 3-D surfaces or their sub-elements during said visibility computation.

(‘047 Patent, col. 27:66-28:17.) The claim language refers to pre-processing the visibility of surfaces (or parts of surfaces) before traditional rendering (“prior to a visibility computation”). The claimed method creates a set of fixed, imaginary planes onto which the surfaces or parts of surfaces of the objects in the scene are projected (“projections”), imposes on each plane an imaginary grid consisting of abstract individual cells (“grid cells”), creates a data structure that stores depth data for each abstraction (“stored data associated with the grid cells”), and then compares the data for a given surface with the data for the corresponding grid cell to determine whether that surface is totally invisible or totally visible. After such pre-processing, an application program can identify which surfaces are totally invisible or totally visible from a given viewpoint group by consulting the data structures associated with the viewpoint group. The application program can then “ignore” (‘047 Patent claim 1), “skip” (‘047 Patent claims 11 and 12), or “exempt” (‘679 Patent claim 1) totally invisible surfaces when sending data to the GPU for rendering an image. Alternatively, the program could send data for totally visible (“always unoccluded”) surfaces to the GPU for rendering but could “ignore,” “exempt,” “skip,” or otherwise direct the GPU not to perform the z-buffering tests for such surfaces. (‘679 Patent claim 1 and ‘047 Patent claims 1, 46.)

The written description likewise describes pre-processing surfaces and for each viewpoint group: classifying the surfaces into categories, creating data structures storing that classification, and then using the stored data to exempt totally visible and totally invisible surfaces from further visibility computations. (*See* ‘679 Patent col. 14:13-25 [“In subsequent rounds of computations, where the form factors between patches have to be re-evaluated, the fuzzy projection computations are not repeated. The totally visible/non-hiding and the totally invisible patches of each-viewpoint can be simply retrieved from the stored information”].)

The overarching goal of the patents is to “reduc[e] the complexity of visibility calculations required for the production of multi-dimensional computer generated images.” (‘679 Patent col. 28:25-27.) The parties have agreed that this means “decreasing the number of computational operations to perform visibility computations.” (Dkt. No. 64 at 5.) By analyzing a scene from a viewpoint or group of viewpoints at the outset and creating data structures reflecting the visibility of surfaces or portions of surfaces rather than individual pixels or groups of pixels, the claimed method was designed to reduce the portions of a scene that had to be subjected to the full-blown visibility and rendering computations that were known and used in the prior art.

<u>CLAIM CONSTRUCTION</u>

For purposes of claim construction, the sole dispute left between the parties is whether the claim terms require a 3-part categorization (*i.e.*, totally hidden, totally visible, and remaining) or whether they are satisfied by a 2-part determination only (*i.e.*, totally hidden vs. remaining, totally visible vs. remaining, or totally visible vs. totally hidden).

Both Group 3 and Group 4 claims concern categorization of surfaces for pre-processing. The parties dispute whether the claims require a three-part determination or a two-part determination. The parties competing constructions follow, with the differences in bold and italics:

Term	**Fuzzysharp's Proposed Construction**	**Intel's Proposed Construction**
Group 3 "determining which of said at least one of 3-D surfaces or their sub-elements is always invisible or always visible" ('047 claim 1)	"determining which of the 3-D surfaces are ***either*** hidden or visible ***from a single viewpoint, or a group of viewpoints***."	"determining whether the 3-D surfaces are totally[3] hidden or totally visible"
Group 4 "determining for said viewpoint, whether each said selected surface is (a) an always unoccluded surface, an always hidden surface, or a remaining surface; or (b) an always unoccluded surface, or a remaining surface; or (c) an always hidden surface, or a remaining surface;" ('679 claim 1)	"for each surface and from a predetermined viewpoint, ***carrying out one of the following groups of tests in (a), or (b), or (c) to determine, respectively***: (a) ***which single one of the following categories the surface falls***: a totally visible surface, a totally hidden surface, or a surface that is neither totally visible or totally hidden; (b) ***which single one of the following categories the surface falls***: a totally visible surface or a surface that is neither totally visible or totally hidden; or (c) ***which single one of the following categories the surface falls***: a totally hidden surface or a surface that is neither totally visible or totally hidden."	"for each surface and from a predetermined viewpoint: (a) ***determining whether the surface is*** a totally visible surface, a totally hidden surface, or a surface that is neither totally visible or totally hidden; ***or*** (b) ***determining whether the surface is*** a totally visible surface or a surface that is neither totally visible or totally hidden; or (c) ***determining whether the surface is*** a totally hidden surface or a surface that is neither totally visible or totally hidden."

[3] Although Fuzzysharp's proposed construction for claim 1 of the '047 patent does not indicate that the terms "hidden" and "visible" mean "totally hidden" and "totally visible," the parties have agreed that they do. (Dkt. No. 64 at 1-2.)

Thus, when Claim 1 of the '679 Patent recites:

> …for selected ones of said surfaces, determining for said viewpoint whether
> each said selected surface is
> (a) an always unoccluded surface, an always hidden surface, or a remaining
> surface; ***or***
> (b) an always unoccluded surface, or a remaining surface; ***or***
> (c) an always hidden surface, or a remaining surface; …

('679 Patent, col. 28:33-39), the parties agree that the italicized "or's" are disjunctive, such that only one of the three sets of tests ((a), (b), and (c)) need be performed. Their dispute lies within each of (a), (b), and (c).

Under Intel's construction, the method of the patents pre-processes the scene to determine from a viewpoint or group of viewpoints which surfaces of the scene are totally hidden, which are totally visible, and which are "remaining." Intel thus proposes that "remaining" means "neither totally hidden nor totally visible" wherever it is used in the patents. Using those data structures, a program may then avoid sending the totally hidden surfaces to the graphics processing unit ("GPU") for rendering, since a totally hidden surface need not be rendered at all. Similarly, the program can send the totally visible surfaces to the GPU, but instruct that no further visibility computations need to be performed to render the surface. Thus, the method of the patents tests for two categories of surfaces, totally hidden or totally visible, and any surface not meeting either of those categories falls into a third, "remaining surfaces." The remaining surfaces are those surfaces that will need additional visibility computations performed by the GPU prior to or during rendering. According to the patents, this categorization of surfaces reduces the number of visibility computations that the GPU must perform, leading to an overall gain in system efficiency.

According to Fuzzysharp, Intel's interpretation of the term "remaining" is inaccurate and leads to confusion. The patents describe different possible operations that test whether surfaces are totally visible or totally invisible. The simpler operations described test only for one type of surface. For example, '679 Patent, Claim 1(c) tests only for hidden surfaces, and under that test a surface is either totally hidden or not. Under this test, Fuzzysharp contends, the "not" is the "remaining surface." Likewise, the operation described in the '679 Patent, Claim 1(b) is to test for totally visible (or unoccluded) surfaces. The nature of this test is that, if the surface fails the test for being visible, then it is classified as being a "remaining" surface, in this case one that can be described as either: (1) totally hidden; or (2) neither totally hidden nor totally visible. Under '679 Patent Claim 1(a) tests are carried out for both totally visible *and* totally hidden surfaces, and failing both tests also means that the surface is "remaining." However, in this scenario "remaining surfaces" would *only* be neither totally hidden nor totally visible. Fuzzysharp further contends that the '047 Patent, Claim 1 describes only two simple operations: a test for always hidden (invisible) surfaces *or* always visible surfaces; those surfaces failing the test used are classified as remaining surfaces. In Fuzzysharp's reading of the claim language, a "remaining" surface is a simply a by-product of steps for determining the information necessary to simplify computations, and therefore means different things depending upon the test used.

I. Claim Language

Unlike Fuzzysharp's proposed construction, Intel's construction gives meaning to the limitations in the claims themselves. First, looking at the disputed portion of the claim in context supports giving a single, consistent meaning to the term "remaining surface" as a category distinct from both "hidden" and "visible/unoccluded." The language of Claim 1 recites, in part:

> …determining for said viewpoint whether each said selected surface is
> (a) an always unoccluded surface, an always hidden surface, or a remaining surface; or
> (b) an always unoccluded surface, or a remaining surface; or
> (c) an always hidden surface, or a remaining surface;
> *wherein said remaining surface is a surface which is unable to be determined with certainty as to whether it is either unoccluded or hidden*

('679 Patent, col. 28:33-42, emphasis added.) Thus, the "wherein" clause is written to apply to *each* of (a), (b), and (c), and to provide a definition of a "remaining surface" for all, not just a definition that applies only to (a). When a "wherein" clause "expresses the inventive discovery," rather than just stating a result of certain limitations, it is definitional. *See Griffin v. Bertina*, 285 F.3d 1029, 1034 (Fed.Cir. 2002); *Intergraph Hardware Technologies Co. v. Toshiba Corp.*, 508 F. Supp. 2d 752, 769 (N.D. Cal. 2007) ("wherein" clause stated limitations that were part of construction rather than expressing result of limitation).

The other limitations of Claim 1 are consistent with Intel's reading. First, Claim 1 requires "exempting from said occlusion or invisibility relationship computation those surfaces which are *either* always unoccluded *or* always hidden." ('679 Patent col. 28:43-45, emphasis added). The "exempting" limitation can be met only if a surface has been tested sufficiently to determine whether it is always visible or always hidden, rather than testing for only one of these categories, as Fuzzysharp suggests. Second, Claim 1 defines "*said remaining surfaces*" in the "wherein" clause and then goes on to require "maintaining a record of *said remaining surfaces*." ('679 Patent col. 28:40-42, 46.) The "maintaining" limitation cannot be met unless surfaces have been tested sufficiently to determine those that are *neither* always visible nor always hidden and thus fall into the third category, "remaining," as to which a record is to be maintained.

Fuzzysharp's proposed construction makes '679 Patent Claim 1 clauses (b) and (c) redundant of clause (a). Fuzzysharp proposes that '679 Patent Claim 1(a) be construed to mean carrying out a test to determine into which ***single*** one of the following categories the surface falls: a totally visible surface, ***or*** a totally hidden surface, ***or*** a surface that is neither totally visible or totally hidden. In other words, a test under Claim 1(a) tests for only one condition, with a binary result (*i.e.*, meets category or does not meet category). Under that construction, however, clauses (b) and (c) would have no meaning, because they would not cover a case not already expressed in clause (a).

Intel's construction gives effect to all three clauses since clause (a) tests for all three categories (totally visible, totally hidden, and "remaining"), clause (b) tests for two of the three categories (totally visible and "remaining," with all others being totally hidden), and clause (c) tests for a different set of two of the three (totally invisible and "remaining," with all others being wholly visible). Thus, Intel's construction fits with the language of '679 Patent Claim 1 itself while Fuzzysharp's does not.

Though the '047 Patent claims do not use the term "remaining surfaces," the same categorization is implicit because those claims simply require determining the totally hidden and totally visible surfaces and then exempting *both* those categories of surfaces from future visibility computations by the GPU. In both cases, the GPU must then perform visibility computations only on the "remaining surfaces." The clause in dispute requires "determining which of said at least one of 3-D surfaces or their sub elements is always invisible or always visible." But the final step requires "ignoring said determined at least one of the 3-D surfaces or their sub-elements during said visibility computations." ('047 Patent, col. 28:9-15.) The claim thus requires testing to determine which surfaces are "always invisible or always visible" so that both wholly hidden and wholly visible surfaces are excluded. Read in context, the "or" does not suggest that only one test need be performed; it instead signifies that a surface cannot be both "always invisible" and "always visible."

In sum, in order for the claim language itself to be internally consistent, the language in both the '679 Patent, Claim 1, and the '047 Patent, Claim 1 require a construction that means the method can test in such a way as to result in three categories of surfaces, not just two.

II. Patent Specification

Fuzzysharp's position is, essentially, that reading the patent to require determining both "always visible" and "always invisible" for all surfaces would not result in the reduction of visibility-related computations that is part of the goal of the patent, as expressed in the Summary of Invention for both patents. ('679 Patent, col. 2: 22-65, '047 Patent, col. 2:22-46.) However, Intel correctly notes that the patent specification in the '679 Patent itself states:

> In the first round of computations, all the fuzzy projection and normal hemicube computations for every receiving patch are carried out as required. ***During these computations, the emission patches at each group/viewpoint are classified into three groups: totally visible/non-hiding, totally invisible, and the remainder.*** This classification information is stored using data structures such as arrays or linked lists.

('679 Patent, col. 14:13-19, emphasis added; *see also* '047 Patent ,col. 13:46-50 [same]). Elsewhere, the specification similarly describes testing for both total invisibility and total visibility, and all others into a "remaining" category:

> "[a]fter the filtering off of totally invisible patches by the invisibility technique, the visibility technique detects totally visible and non-hiding patches. … The remaining patches are not totally invisible. They are also not totally visible/non-hiding."

(‘679 Patent, col. 14:59-65; *see also* ‘047 Patent, col. 14:28-29 [same].).) Thus the specification supports a construction requiring that the claims will, at least in some cases, put surfaces into three categories. Contrary to Fuzzysharp’s argument, such a construction is also consistent with the goal of eliminating a full visibility computation for surfaces already known to be wholly visible or wholly hidden. Thus the specification supports Intel’s construction reading the claim language in both the ‘679 Patent, Claim 1, and the ‘047 Patent, Claim 1 to result in a determination of three categories of surfaces.

III. PROSECUTION HISTORY

While not necessary to the Court’s construction determination, Intel’s proposed construction is consistent with the prosecution history stated in the patents. The patent examiner initially rejected the claim language in the ‘047 Patent which stated “determining which of the 3-D surfaces or their sub-elements are always invisible or always visible.” (Dkt. No. 66-2 to 66-6 [Declaration of Victoria Q. Smith], Exh. C [‘047 Patent File History, Application] at 48.) The patent examiner found that the claim was obvious in light of the prior art in *Kadota*, which disclosed a determination whether 3-D surfaces were always visible or invisible from a viewpoint. The rejection stated:

> ***As per independent claim 1, Kadota discloses determining 3D surfaces that are always visible or invisible*** to a viewpoint by projection (abstract) and treating those surfaces differently from remaining surfaces …, it would have been obvious to one of ordinary skill in the art at the time of invention to use the disclosure of Kadota because he teaches transmitting and processing predetermined projection coordinates to a hidden surface processor ***which evaluates both visible and invisible surfaces***, transmits the visible surfaces and ignores the invisible surfaces.

(Smith Dec., Exh. E [‘047 Patent File History, September 22, 2000 Rejection] at 2, emphasis added.)

Previously, the Patent and Trademark Office had allowed the ‘679 Patent, the Reasons for Allowance having stated:

> the applicant claims sorting surfaces as visible, hidden or remaining (either visible or nonvisible) and processing the remaining surfaces to perform and

United States District Court
Northern District of California

> reduce visible/hidden surface computations, which is not disclosed in the prior art of record.

(Smith Dec., Exh. D [‘679 Patent File History, November 15, 1999 Notice of Allowance] at 2.) Fuzzysharp does not address this prosecution history or offer any contradictory arguments.

Thus, based on the prosecution history offered by Intel, the Patent and Trademark Office appears to have considered the three categories of surfaces to be critical in distinguishing the patents over prior art.

THE COURT’S CONSTRUCTION:

Based upon the foregoing, the Court interprets the claims at issue as follows:

Term	**Construction**
Group 3(‘047 claim 1) “determining which of said at least one of 3-D surfaces or their sub-elements is always invisible or always visible”	“determining whether the 3-D surfaces are totally hidden or totally visible”
Group 4 (‘679 claim 1) “determining for said viewpoint, whether each said selected surface is (a) an always unoccluded surface, an always hidden surface, or a remaining surface; or (b) an always unoccluded surface, or a remaining surface; or (c) an always hidden surface, or a remaining surface;”	“for each surface and from a predetermined viewpoint: (a) determining whether the surface is a totally visible surface, a totally hidden surface, or a surface that is neither totally visible or totally hidden; or (b) determining whether the surface is a totally visible surface or a surface that is neither totally visible or totally hidden; or (c) determining whether the surface is a totally hidden surface or a surface that is neither totally visible or totally hidden.”

//

//

SUMMARY JUDGMENT

Intel brings its motion for summary judgment on the grounds that, pursuant to 35 U.S.C. § 101, the Asserted Claims are not patent-eligible. Intel argues that the Asserted Claims set forth an abstract idea without adding more than an instruction to "apply it," or recite an abstract idea without requiring any application containing an "inventive concept," or both. Whether the claims satisfy 35 U.S.C. § 101's criteria for patent-eligible subject matter is a question of law for the Court. *Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1333 (Fed. Cir. 2012). The Federal Circuit has held that "any attack on an issued patent based on a challenge to the eligibility of the subject matter must be proven by clear and convincing evidence." *Ultramercial, Inc. v. Hulu, LLC*, 722 F.3d 1335, 1342 (Fed. Cir. 2013).

Here, there are no disputed issues of fact raised by the parties.[4] The parties agreed on constructions for all of the claim terms in dispute but two. Those two terms were the subject of claim construction briefing and argument, and their constructions are set forth above, in the first part of this Order.

I. LEGAL FRAMEWORK

In analyzing patentability under Section 101, a court must first identify whether the claimed invention fits within one of the four statutory classes set forth in the statute. Section 101 specifies four broad, independent categories of inventions or discoveries that are patent eligible: "process, machine, manufacture or composition of matter." 35 U.S.C. § 101. Second, a court must assess whether the claimed invention falls into any of the three specific exceptions established by Supreme Court precedent: "laws of nature, physical phenomena, and abstract ideas." *Diamond v Chakrabarty*, 447 U.S. 303, 309 (1980). These exceptions have grown out of the Supreme Court's

[4] Intel initially brought this motion as one for judgment on the pleadings. (Dkt. No. 28.) Based upon the arguments made in opposition by Fuzzysharp, the Court converted the motion to one for summary judgment and permitted the parties to file supplemental briefing and evidence, and specifically directed that the parties identify any terms or phrases that required a claim construction prior to a decision on summary judgment. (Dkt. No. 45.)

"concern that patent law not inhibit further discovery by improperly tying up the future use of laws of nature," natural phenomena, and abstract ideas. *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 132 S. Ct. 1289, 1301 (2012). "A principle, in the abstract, is a fundamental truth…[which] cannot be patented." *Gottschalk v. Benson,* 409 U.S. 63, 67 (1972) (internal citations and quotations omitted) ("*Benson*"). "Phenomena of nature, though just discovered, mental processes, and abstract intellectual concepts are not patentable, as they are the basic tools of scientific and technological work." *Id.* Included within the exception for abstract ideas are such concepts as mathematical formulae. *Id.* at 68.

A court must then determine whether the patent claims "pose any risk of preempting an abstract idea." *CLS Bank Int'l v. Alice Corp. Pty. Ltd.*, 717 F.3d 1269, 1282 (Fed. Cir. 2013). Even if the patent claims rely on an abstract concept such as a mathematical formula, the "balance of the claim can be evaluated to determine whether it contains additional substantive limitations that narrow, confine, or otherwise tie down the claim so that, in practical terms, it does not cover the full abstract idea itself." *Id.* at 1282 (citing *Mayo*, 132 S.Ct. at 1300; *Bilski v. Kappos*, 130 S. Ct. 3218, 3231 (2010); and *Diehr,* 450 U.S. at 187).

A. Supreme Court Authorities

Over time, the Supreme Court has set forth a number of different factors to consider in distinguishing a patentable invention from an unpatentable abstract concept. Six cases are particularly instructive here. First, in *Benson*, the Supreme Court held that a mathematical formula for converting binary-coded decimal numerals into pure binary code *was not* patentable because the process claimed was "so abstract and sweeping" that it could cover usage in a variety of operational settings and could be performed through any existing or future machinery, or without machinery at all. *Benson,* 409 U.S. at 67. The fact that the process or method is carried out with the use of common computer components, such as computer memory, does not render an otherwise abstract concept patentable and processes that "can be carried out in existing computers long in use, no new machinery being necessary" are ineligible for patent protection. *Id.* at 67.

Following thereafter, in *Parker v. Flook*, 437 U.S. 584 (1978) ("*Flook*"), the Supreme Court held that a patent which described a mathematical formula for calculating alarm limit values *was* *not* patentable due to the abstract nature of the subject matter. There, the patentee sought to distinguish *Benson* by arguing that, although there were a number of "post-solution" uses for the formula, the patent included specific "post-solution" activity to make the formula useful in the petrochemical and oil-refining industries. The Supreme Court rejected this distinction, stating that

> [t]he notion that post-solution activity, no matter how conventional or obvious in itself, can transform an unpatentable principle into a patentable process exalts form over substance. A competent draftsman could attach some form of post-solution activity to almost any mathematical formula; the Pythagorean theorem would not have been patentable, or partially patentable, because a patent application contained a final step indicating that the formula, when solved, could be usefully applied to existing surveying techniques.

Flook, 437 U.S. at 590 (1978). Thus, while a claimed invention does not lose patent eligibility simply because it relies on a phenomenon of nature or mathematical formula, such a phenomenon or formula cannot, on its own, be patentable without "some other inventive concept in its application." *Id.* at 594.

Subsequently, by contrast, the Supreme Court found patentable under Section 101 a process combining an abstract formula with practical applications and limitations in *Diamond v. Diehr*, 450 U.S. 175 (1981) ("*Diehr*"). There, the patent claimed a process for curing synthetic rubber that, while involving an abstract mathematical formula, used that formula in combination with a novel set of steps to be followed in a particular application. The overall process was patentable because it integrated the equation into the rubber curing process as a whole, using constant temperature readings inside the mold, and making repeated recalculations to determine the optimum time to open the rubber press. *Id.* The Court held that patent avoided the abstractness exception to patent eligibility since the patentees did not "seek to pre-empt the use of that equation. Rather, they seek only to foreclose from others the use of that equation *in conjunction with all of the other steps* in their claimed process." *Id.* at 187 (emphasis added). Thus an abstract idea, law of nature, or mathematical formula could not be patented but "an *application* of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection." *Id.*

In its 2010 *Bilski* decision, the Supreme Court reiterated these principles. *Bilski*, 130 S. Ct. at 3231. The Court held that the Federal Circuit's "machine-or-transformation test," while not dispositive of patent eligibility, provides a "useful tool" for determining whether a claimed method or process is patent eligible. *Bilski*, 130 S. Ct. at 3226-27. Under that formulation, a method or process is patent eligible when: "(1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing." *Id.* at 3225 (quoting *In re Bilski*, 545 F.3d 943, 954 (Fed. Cir. 2008) (*en banc*)). In *Bilski*, the Court found that claims describing the concept of hedging and a mathematical formula for hedging against risk in price fluctuations for buyers and sellers of commodities were not patentable. The Court held that "hedging" was a basic economic concept, and reducing the concept to a mathematical formula did not take it out of the realm of abstract ideas. *Id.* Likewise, adding claims that simply applied the formula to particular markets was not enough to convert the abstract concept into a patentable one, just as adding a field of use or common post-solution components was not sufficient in *Flook*. *Id.*

Most recently, the Supreme Court held that claims were too abstract, and not patentable under section 101, where "the claimed processes (apart from the natural laws themselves) involve well-understood, routine, conventional activity previously engaged in by researchers in the field…." *Mayo*, 132 S. Ct. at 1294. The Court held that a claim must apply an abstract idea in a way that reflects an "inventive concept," which it illustrated by distinguishing its decisions in *Flook* and *Diehr*. In *Flook*, the claimed method used a computer to calculate alarm-limit values to signal dangers in operating a catalytic converter, but the process did "nothing other than 'provid[e] an unpatentable formula for computing [the] updated alarm limit." *Mayo*, 132 S. Ct. at 1299 (quoting *Flook*, 437 U.S. at 586) (internal quotations and brackets omitted). All the other steps—the chemical processes involved in catalytic conversion, the monitoring of chemical process variables, the use of alarm limits to signal danger, and the adjusting of those limits through the use of computers for "automatic monitoring-alarming"—were "'well known.'" *Id.* at 1299 (quoting *Flook*, 437 U.S. at 594). Thus, "putting the [abstract] formula to the side, there was no 'inventive concept' in the claimed application of the formula." *Id.* In *Diehr*, by contrast, the "other steps" added something "that transformed the process into an inventive application of the formula." *Mayo*, 132 S Ct. at 1299 (process was patentable because the other steps, and the combination of them, were not "obvious, already in use, or purely conventional").

B. Recent Federal Circuit Decisions

The Federal Circuit has applied these principles in a recent cases involving computing technologies. In one, the Federal Circuit found unpatentable a claimed method which disclosed a number of computer components needed to carry it out. *Accenture Global Servs., GmbH v. Guidewire Software, Inc.*, 728 F.3d 1336, 1343 (Fed. Cir. 2013) (components required included "CPU, ROM, RAM, I/O Adapter, Communication Adapter, Display Adapter, and a User Interface Adapter"). The listed computer components constituted what was described in the patent as a "representative hardware environment." *Id.* The Federal Circuit found such components insufficient to provide meaningful limitations on the abstract concept of the patent itself. *Id.* "[S]imply implementing an abstract concept on a computer, without meaningful limitations to that concept, does not transform a patent-ineligible claim into a patent-eligible one." *Id.* at 1345 (citing *Bancorp*, 687 F.3d at 1280). "[T]he complexity of the implementing software or the level of detail in the specification does not transform a claim reciting only an abstract concept into a patent-eligible system or method." *Id.*

Accenture echoed other recent Federal Circuit decisions finding that simply adding a storage or calculation step performed by a digital computer component did not limit an otherwise abstract concept sufficiently to make it patentable. *See Bancorp Servs., L.L.C. v. Sun Life Assur. Co. of Canada (U.S.)*, 687 F.3d 1266, 1278 (Fed. Cir. 2012) (computer and "high density removable storage means … such as a compact disc" required by patent claims used only for basic computing functions and repetitive calculations did not impose meaningful limits on the scope of those claims to make them patentable); *Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1333 (Fed. Cir. 2012) ("simply adding a 'computer aided' limitation to a claim covering an abstract concept, without more, is insufficient to render the claim patent eligible"); *Cybersource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1376 (Fed. Cir. 2011) (added computer storage step naming particular type of storage component does not render otherwise unpatentable process patent-eligible) (citing *Benson*, 409 U.S. at 73-74).

By contrast, the Federal Circuit in *Ultramercial* found claims concerning a method for distributing media products over the Internet to be patent-eligible. *Ultramercial*, 722 F.3d at 1352-

53. There, the claims required steps to be performed through an "extensive computer interface" involving "eleven separate and specific steps with many limitations and sub-steps in each category." *Id.* There, the Federal Circuit reiterated that "[m]ere reference to a general purpose computer will not save a method claim from being deemed too abstract to be patent eligible." *Id.* at 1348 (citing *Bilski*, 130 S.Ct. at 3227). However, it held that the requirements of the patent claims at issue in *Ultramercial* meaningfully limited the abstract formulae and algorithms at the core of the claims so that the claimed invention was not "so manifestly abstract as to override the statutory language of section 101." *Id.* (quoting *Research Corp. Technologies, Inc. v. Microsoft Corp.*, 627 F.3d 859, 869 (Fed. Cir. 2010) [upholding claimed method for rendering a halftone image by manipulation of computer data and output]).

It is with these principles and authorities in mind that the Court turns to the question of patent eligibility presented here.

II. APPLICATION TO PATENTS-IN-SUIT

The Court agrees that the claims here do not state patentable subject matter. The patents describe their invention as "an improved method for performing visibility calculations" in three-dimensional graphics. ('047 Patent, col. 2:23-24; '679 Patent ('679 Patent), col. 2:19-20; patent Titles.) Visibility calculations are used for "visible surface detection," which is "one of the most basic operations" in 3D graphics. ('047 Patent, col. 1:18-21; '679 Patent, col. 1:16-18.) The claimed methods of the patents shorten the calculations by *identifying* the parts of the surfaces in a 3-D representation that are always visible or always invisible from a set of viewpoints, and then *skipping* the calculations for that part in further rendering of the image. The '047 and '679 preambles claim methods of reducing "visibility calculations" (or "computations") "in 3-D computer graphics" ('047) and "required for the production of multidimensional computer generated images" ('679). As agreed by the parties, these phrases mean: "decreasing the number of computational operations required to perform prior art visibility computations" "in the field of three-dimensional computer graphics" ('047) or "to produce a 3-D image." ('679). (Dkt.No. 64 at 5.)[5]

All of the Asserted Claims adhere to the same basic format as independent claim 1 of the '047 Patent. First, the claims include a preamble that states that the method is for reducing the visibility related computations calculations in a certain field of use (such as "3-D computer graphics"), followed by method steps to perform the reduction. The method steps have three components:

(1) mathematical manipulations of data, such as "identifying," "comparing," or "determining;"

(2) a reducing step ("ignoring" or "exempting" or "skipping" data or calculations); and

(3) an indication of where these steps fit into the visibility computations.

Finally, there are a few references to generic computing structures (*e.g.*, "data," a "record," or a "computer"). In sum, the patents claim a mathematical principle for eliminating or simplifying portions of otherwise well-known and established computations by pre-determining a portion of the calculations, and then specifying that a portion of the calculations can be skipped or omitted going forward. The claims specify no physical output or programming result. Rather, in every claim, the last method step recites either ignoring part of the visibility computations or completing such calculations without the unnecessary steps.

[5] The parties have defined "visibility calculations," in relevant part, as "rendering computations that determine which pixels … of 3D surface(s) are visible," and thus such calculations already imply the production of a 3D image. (Dkt. No. 64 at 2.) The parties also agreed on the following definitions of terms:

"projections" and "projection planes" refer to "representations of a 3D object's surfaces on an imaginary plane";

"grid cells" are an "imaginary lattice structure";

a "bounding volume" is "the smallest imaginary right quadrangular prism just enclosing the 3-D object's surfaces"; and

"surfaces" are simply "surfaces" of 3D objects that are part of a scene.

(Dkt. No. 64 at 1-3.)

A. Lack of Meaningful Limitation To A Specific Application/Inventive Concept

The claims do not limit the mathematical formula to any specific use beyond the broad field of invention of "computer graphics." ('047 Patent, col. 1:11-16; '679 Patent, col. 1:8-13; '679 and '047 Patents, Abstract [the disclosed method operates "in 3D graphics systems"].) Simply limiting the use of an abstract formula to a particular field is insufficient to avoid the prohibition on patenting abstract ideas. *Bilski*, 130 S.Ct. at 3230; *Mayo*, 132 S.Ct. at 1301.

Similarly, the claims only state a reduction in computations, with no specific output. The claims cover any application for which such computations might be used. Any equations having the same "mathematical relationship of entities" fall within the invention regardless of the physical properties or variables to which they are applied. ('047 Patent, col.4:6:11; '679 Patent, col.4:36-40.) Just as the method of converting binary numbers in *Benson* would cover all computer applications of that algorithm, the claims at issue here would cover all applications of their algorithm in the field of "3D computer graphics." Indeed, the patent specification indicates that the applications of the formula are numerous and would include any process that requires visible surface detection, such as "radiosity calculations to compute the energy interactions between surfaces," "Hidden Surface Computations" for "computer animation, flight simulation or dynamic graphics," "Ray Tracing Computations," "Computer Vision," "virtual reality applications," and the "processing and display of scientific data such as energy spectra data." ('047 Patent, col. 1:20-22, 15:33-41, 15:58-16:19, 16:21-59, 18:33-46, 18:46-48, 18:56-59.)

B. Conventional Post-Solution Activity

Moreover, the claims do not add any steps other than conventional "post-solution" activity to the abstract formula described, making them unpatentable as stated in *Flook* and *Mayo*. As the patent specifications explain, "[v]isible surface detection is one of the most basic operations in 3D graphics" for generating 3D images. ('047 Patent, col. 1:18-21, 2:25-41; '679 Patent, col. 1:16-18, 2:20-27.) The patents purport to make surface detection computations faster by computing the "surface elements obviously visible or invisible to each other," and ignoring those elements, thereby "reducing the visibility related computations." ('047 Patent, col. 1:40-43, 2:21-44; '679 Patent, col. 1:38-40, 2:18-28.) The claims confirm that conventional visibility computations follow the method steps. Claim 1 of the ‘679 Patent recites that the entire method is performed “prior to an occlusion or invisibility relationship computation (*known per se*) being carried out.” (‘679 Patent, col. 28:31 (emphasis added).) The parties agree that this refers to “prior art types of rendering computations that determine which pixels or groups of pixels of 3D surface(s) are visible, which include z-buffering and other visibility tests performed during rendering.” (Dkt. No. 64 at 2.) The claims simply recite the steps of determining what part of well-known visibility computations may be omitted, omitting them, and carrying out the remaining computations as before. They do not alter the remainder of the rendering process of which those computations are a part.

Thus, the steps that follow are visibility computations that would be performed even if the claimed method were not used. There is nothing in the patents’ claims or specifications that adds to the formula steps, or a combination of steps, that would transform the otherwise abstract formula into an “inventive concept.” *See Mayo*, 132 S Ct. at 1299. As in *Flook*, the claims are drawn to a mathematical formula and merely append conventional activity, which cannot supply the “particular application” needed to impart patentability. *Mayo*, 132 S. Ct. at 1299 (citing *Flook*).

C. Inclusion of General Computer Components

Finally, the recitation that the method is to be used on conventional computer components does not make the abstract formula patentable. The parties’ constructions of the terms describing the physical components on which the patent method is practiced confirm that those structures are simply generic computer components. The physical components recited in the claims include “computer,” “computer storage,” and “z-buffers.” “Computer” was given no construction, while “computer storage” is agreed to mean simply “computer memory.” The term “z-buffer” was agreed to mean “a data structure in memory that is used to store depth data,” again a generic term. (Dkt. No. 64 at 6.)

The inclusion of these computer components does not impose a limitation sufficient to take the patents out of the realm of the abstract. As noted by Defendant, five of the claims here — claims 1, 4, and 5 of the ‘679 Patent and claims 1 and 12 of the ‘047 Patent— were the subject of prior litigation in which they were found to fail the machine-or-transformation test. *Fuzzysharp Techs., Inc. v. 3D Labs, Inc.*, 2009 WL 4899215, at *5 (N.D. Cal. Dec. 11, 2009). There the district court, examining some of the exact claims at issue here, found that the claimed sequence of "'identifying,' 'comparing,' 'determining,' and 'ignoring' data" "may be" performed on a computer but does not require "any *particular* computer." *Id.* at *4. Nor did such elements as a "method of reducing the [ ]visibility related computations in 3-D graphics," "computer storage," "using a data structure in a computer," and "projecting 3D images 'on a computer screen'" "impose any meaningful limit on the claim scope" because it merely "serve[d] to perform the computation." *Id.* at *5 n.3. On appeal, the Federal Circuit found:

> [t]he references to a computer in claim 12 impose only two limitations: the machine must be able to compute, and it must be able to store data. Those functions are essentially synonymous with the term "computer" and thus add little or nothing to simply claiming the use of a general purpose computer. The recitation of computer functions in the claim thus does not confine the preemptive effect of the claim because the underlying method has "no substantial practical application except in connection with a digital computer."

Fuzzysharp Technologies Inc. v. 3DLabs Inc., Ltd., 447 F. App'x 182, 185 (Fed. Cir. 2011), reh'g denied (Dec. 20, 2011) (quoting *Benson,* 409 U.S. at 71). Thus the Federal Circuit agreed with the district court's conclusion that the claims were drawn to an abstract mathematical formula. *Id.* at 184. However, the district court had relied exclusively on the machine-or-transformation test, which the Supreme Court subsequently held was not a definitive test of patent eligibility for a method claims, but simply "a useful and important clue" to patent-eligibility. *Bilski,* 130 S. Ct. at 3227. Thus, the Federal Circuit vacated and remanded the judgment in the *3DLabs* action to permit further consideration in light of *Bilski* and to allow for claim construction related to eligibility of one unspecified claim there. *Id.* at 186.

The Court agrees with the determination in the *3DLabs* case that the computer components included in the language of the Asserted Claims does not provide a meaningful limitation on the abstract mathematical formula so as to render it patent eligible. The stated components are no more specific than the computer components or computer-aided processing described repeatedly in the case law as insignificant to patentability. As in *Benson, Accenture,* and similar authorities, the specification of the method as requiring use of generic computer memory does not meaningfully limit the broad concept set forth in the claims. *See Benson*, 409 U.S. at 67; *Accenture*, 728 F.3d at

1345-46. Adding a computer storage step to an otherwise unpatentable process, even where the claim names a particular type of storage component like the z-buffer here, does not transform it into a claim that is patent-eligible. *Cybersource*, 654 F.3d at 1375-76 (citing *Benson*, 409 U.S. at 73-74). Here, no claim recites any specific programming or computer component, but only geometric calculations. Performing those calculations on a computer, even if their only practical application is in a computing environment, does not render them patent-eligible. *Id.*

The eight additional Asserted Claims of the ‘047 Patent, not at issue in *3DLabs*, consist of the same method steps as the others (*i.e.*, identifying, comparing and determining visibility data, and then ignoring a set of data for certain surfaces in carrying out standard visibility calculations thereafter). As a result, those claims likewise suffer from the same lack of meaningful limitations on the mathematical formula at the core of the patent claims.

Unlike *Ultramercial*, the claims here suggest purely mathematical efficiencies, and do not require an "extensive computer interface" involving multiple steps and limitations. *Ultramercial*, 722 F.3d at 1352-53. Nor do the claims in the patents-in-suit implicate particular physical elements beyond a generic computer environment for carrying out their steps. *Cf. Research Corp. Technologies*, 627 F.3d at 869 (invention describing halftone rendering process in which digital data processor was utilized, along with "high contrast film," "film printer," as well as generic computer devices such as "memory" and "printer and display devices," was not so abstract as to be lose patent eligibility).

III. CONCLUSION

Based upon the foregoing, Intel's Motion for Summary Judgment for patent ineligibility of the Asserted Claims under Section 101 is **GRANTED**.

Intel is directed to submit a proposed form of judgment forthwith.

This Order terminates Docket No. 28.

IT IS SO ORDERED.

Date:November 6, 2013

Yvonne Gonzalez Rogers

YVONNE GONZALEZ ROGERS
UNITED STATES DISTRICT COURT JUDGE